



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/812,643	03/29/2004	Adam P. Donlin	X-1484 US	3407
24309	7590	07/30/2008	EXAMINER	
XILINX, INC			BROWN, MICHAEL J	
ATTN: LEGAL DEPARTMENT				
2100 LOGIC DR			ART UNIT	PAPER NUMBER
SAN JOSE, CA 95124			2116	
			MAIL DATE	DELIVERY MODE
			07/30/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/812,643	DONLIN ET AL.	
	Examiner	Art Unit	
	Michael J. Brown	2116	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 08 May 2008.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-30 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-13, 16 and 18-30 is/are rejected.
 7) Claim(s) 14, 15 and 17 is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 29 March 2004 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____ .
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)	5) <input type="checkbox"/> Notice of Informal Patent Application
Paper No(s)/Mail Date _____.	6) <input type="checkbox"/> Other: _____ .

DETAILED ACTION

Allowable Subject Matter

1. Claims 14, 15, and 17 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

2. Claims 1-13, 16, and 18-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Muhlestein et al.[Muhlestein](US PGPub 2003/0191810) in view of Bhesania et al.[Bhesania](US PGPub 2002/0122056).

As to claim 1, Muhlestein discloses a method for providing access to data in a programmable integrated circuit(IC)(field programmable gate array(FPGA); see

paragraph 0051, lines 7-8), comprising maintaining a hierarchy of directories and files(hierarchy of directories and files; see paragraph 0060, lines 3-4) in a virtual file system(virtual filer(VF2), see Fig. 4) that is registered with an operating system(storage operating system 300, see Fig. 4), wherein the directories and files are associated with resources of the programmable IC(field programmable gate array(FPGA); see paragraph 0051, lines 7-8)(see paragraph 0060, lines 1-7); in response to program calls(incoming request; see paragraph 0090, lines 6-7) to file system routines that reference files associated with resources(storage resource; see paragraph 0096, line 6) of the programmable IC, invoking the virtual file system(see paragraph 0096, lines 3-5 and Step 818, see Fig. 8); and accessing state information(configuration information stored in an IPSpace database; see paragraph 0090, lines17-18) in resources of the programmable IC by the virtual file system(see paragraph 0096, lines 5-6 and Steps 820 and 830, see Fig. 8). However, Muhlestein fails to specifically disclose the method wherein the directories and files have names indicative of resources of the programmable IC.

Bhesania teaches directories and files(information files; see paragraph 0015, lines 10-11) having names indicative of resources(see paragraph 0015, lines 10-12). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Muhlestein's and Bhesania's inventions in order to name Muhlestein's hierarchy of directories and files according to their resources. The motivation to do so would have been to better manage those files, certainly for someone who is accessing the field programmable gate array(FPGA) for the first time.

As to claim 2, Muhlestein discloses the method of claim 1, wherein the resources include configurable logic resources(see paragraph 0060, lines 12-15).

As to claim 3, Muhlestein discloses the method of claim 2, wherein the resources include storage resources(storage resource; see paragraph 0096, line 6).

As to claim 4, Muhlestein discloses the method of claim 1, further comprising reading configuration data from a set of resources of the programmable IC in response to a call to a first file system routine that references a file with which the set of resources is associated(see paragraph 0096, lines 1-6).

As to claim 5, Muhlestein discloses the method of claim 4, further comprising writing configuration data to a set of resources of the programmable IC in response to a call to a second file system routine that references a file with which the set of resources is associated(see paragraph 0096, lines 1-6).

As to claim 6, Muhlestein discloses the method of claim 1, wherein the hierarchy of directories and files includes a directory with a plurality of region files, each region file associated with resources within an area of the PROGRAMMABLE IC associated with the region file(see paragraph 0060).

As to claim 7, Muhlestein discloses the method of claim 1, wherein the hierarchy of directories and files includes an application directory that includes a hierarchy of application sub-directories and application files, wherein each application subdirectory is associated with a subsystem implemented on the programmable IC, and each

application file under an application sub-directory represents resources associated with the subsystem of the application sub-directory(see paragraph 0060).

As to claim 8, Muhlestein discloses the method of claim 5, further comprising associating access permission indicators with selected ones of the directories and files; granting read access to configuration data from a set of configurable resources associated with a file in response to a first state of an access permission indicator associated with the file; and denying read access to configuration data from a set of resources associated with a file in response to a second state of the access permission indicator associated with the file(see paragraph 0096 and Fig. 8).

As to claim 9, Muhlestein discloses the method of claim 8, further comprising granting write access to configuration data from a set of configurable resources associated with a file in response to a third state of an access permission indicator associated with the file; and denying write access to configuration data from a set of resources associated with a file in response to a fourth state of the access permission indicator associated with the file(see paragraph 0096 and Fig. 8).

As to claim 10, Muhlestein discloses the method of claim 1, wherein the hierarchy of directories and files includes a directory with an executable file, the executable file configured to access state information of the programmable IC, and the method further comprising associating access permission indicators with selected ones of the directories and files; granting execution access to an executable file in response to a first state of an access permission indicator associated with the executable file; and denying execution access to an executable file in response to a second state of the

access permission indicator associated with the executable file(see paragraph 0096 and Fig. 8).

As to claim 11, Muhlestein discloses the method of claim 1, wherein the hierarchy of directories and files includes a directory with an executable file, the executable file configured with instructions for interpretation by the virtual file system and instructing access to state data of the programmable IC, and the method further comprising, in response to a request for execution of the executable file, instructing the virtual file system to interpret the instructions of the executable file(see paragraph 0096 and Fig. 8).

As to claim 12, Muhlestein discloses the method of claim 1, wherein the hierarchy of directories and files includes a directory with at least one executable file, the executable file configured with one or more control codes for transferring control to the virtual file system, and the method further comprising, in response to a request for execution of the executable file, transferring control to the virtual file system(see paragraph 0096 and Fig. 8).

As to claim 13, Muhlestein discloses the method of claim 5, wherein the programmable IC is coupled to a node(intermediate network node 150, see Fig. 1) via a network(network cloud 130, see Fig. 1), further comprising exporting information describing the hierarchy of directories and files to the node; and providing network access to the first and second functions of the virtual file system(see paragraph 0096 and Fig. 8).

As to claim 16, Muhlestein discloses the method of claim 1, wherein the hierarchy of directories and files includes a directory with a plurality of region directories and each region directory includes a configuration file and a state file, each configuration file associated with configurable logic resources within an area of the programmable IC associated with the region directory, and each state file associated with storage resources within the area of the programmable IC associated with the region directory(see paragraph 0096 and Fig. 8).

As to claim 18, Muhlestein discloses the method of claim 1, further comprising implementing a processor(processor 202, see Fig. 2) on a programmable IC; and hosting the operating system on the processor(see Fig. 2).

As to claim 19, Muhlestein discloses the method of claim 1, further comprising: interfacing the virtual file system with a configuration controller(intermediate network node 150, see Fig. 1) implemented on the programmable IC; and accessing programmable IC resources via the configuration controller in response to access requests from the virtual file system(see paragraph 0040).

As to claim 20, Muhlestein discloses the method of claim 19, wherein the interfacing step comprises interfacing the virtual file system with a configuration controller via a network(network cloud 130, see Fig. 1).

As to claim 21, Muhlestein discloses the method of claim 1, further comprising writing a configuration file to a sym file handle(file handle; see paragraph 0047, line 12) provided by the virtual file system, wherein the configuration file specifies the hierarchy

of directories and files; and in response to writing of the configuration file, establishing the hierarchy of directories and files by the virtual file system(see paragraph 0047).

As to claim 22, Muhlestein discloses a method for providing access to data in a programmable integrated circuit(IC)(field programmable gate array(FPGA); see paragraph 0051, lines 7-8), comprising maintaining a hierarchy of directories and files(hierarchy of directories and files; see paragraph 0060, lines 3-4) in a virtual file system(virtual filer(VF2), see Fig. 4) that is registered with an operating system(storage operating system 300, see Fig. 4), wherein the directories and files are associated with resources of the programmable IC(see paragraph 0060, lines 1-7); in response to program calls(incoming request; see paragraph 0090, lines 6-7) to file system routines that reference files associated with resources(storage resource; see paragraph 0096, line 6) of the programmable IC, invoking the virtual file system(see paragraph 0096, lines 3-5 and Step 818, see Fig. 8); and accessing by the virtual file system state information(destination IP address and IPspace ID; see paragraph 0090, lines 10-11) in a bitstream file containing state information(configuration information stored in an IPSpace database; see paragraph 0090, lines17-18) of resources of the programmable IC(see paragraph 0096, lines 5-6 and Steps 820 and 830, see Fig. 8). However, Muhlestein fails to specifically disclose the method wherein the directories and files have names indicative of resources of the programmable IC.

Bhesania teaches directories and files(information files; see paragraph 0015, lines 10-11) having names indicative of resources(see paragraph 0015, lines 10-12). It would have been obvious to one of ordinary skill in the art at the time the invention was

made to combine Muhlestein's and Bhesania's inventions in order to name Muhlestein's hierarchy of directories and files according to their resources. The motivation to do so would have been to better manage those files, certainly for someone who is accessing the field programmable gate array(FPGA) for the first time.

As to claim 23, Muhlestein discloses an apparatus for providing access to data in a programmable integrated circuit(IC)(field programmable gate array(FPGA); see paragraph 0051, lines 7-8), comprising means for maintaining a hierarchy of directories and files(hierarchy of directories and files; see paragraph 0060, lines 3-4) in a virtual file system(virtual filer(VF2), see Fig. 4) that is registered with an operating system(storage operating system 300, see Fig. 4), wherein the directories and files are associated with resources of the programmable IC(see paragraph 0060, lines 1-7); and means, responsive to program calls(incoming request; see paragraph 0090, lines 6-7) to file system routines that reference files associated with resources(storage resource; see paragraph 0096, line 6) of the programmable IC, for invoking the virtual file system(see paragraph 0096, lines 3-5 and Step 818, see Fig. 8) and accessing state information(configuration information stored in an IPSpace database; see paragraph 0090, lines17-18) in resources of the programmable IC by the virtual file system(see paragraph 0096, lines 5-6 and Steps 820 and 830, see Fig. 8). However, Muhlestein fails to specifically disclose the apparatus wherein the directories and files have names indicative of resources of the programmable IC.

Bhesania teaches directories and files(information files; see paragraph 0015, lines 10-11) having names indicative of resources(see paragraph 0015, lines 10-12). It

would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Muhlestein's and Bhesania's inventions in order to name Muhlestein's hierarchy of directories and files according to their resources. The motivation to do so would have been to better manage those files, certainly for someone who is accessing the field programmable gate array(FPGA) for the first time.

As to claim 24, Muhlestein discloses an apparatus for providing access to data in a programmable integrated circuit(IC)(field programmable gate array(FPGA); see paragraph 0051, lines 7-8), comprising means for maintaining a hierarchy of directories and files(hierarchy of directories and files; see paragraph 0060, lines 3-4) in a virtual file system(virtual filer(VF2), see Fig. 4) that is registered with an operating system(storage operating system 300, see Fig. 4), wherein the directories and files are associated with resources of the programmable IC(see paragraph 0060, lines 1-7); and means, responsive to program calls(incoming request; see paragraph 0090, lines 6-7) to file system routines that reference files associated with resources(storage resource; see paragraph 0096, line 6) of the programmable IC, for invoking the virtual file system(see paragraph 0096, lines 3-5 and Step 818, see Fig. 8) and accessing by the virtual file system state information(destination IP address and IPspace ID; see paragraph 0090, lines 10-11) in a bitstream file containing state information(configuration information stored in an IPSpace database; see paragraph 0090, lines17-18) of resources of the programmable IC(see paragraph 0096, lines 5-6 and Steps 820 and 830, see Fig. 8). However, Muhlestein fails to specifically disclose the apparatus wherein the directories and files have names indicative of resources of the programmable IC.

Bhesania teaches directories and files(information files; see paragraph 0015, lines 10-11) having names indicative of resources(see paragraph 0015, lines 10-12). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Muhlestein's and Bhesania's inventions in order to name Muhlestein's hierarchy of directories and files according to their resources. The motivation to do so would have been to better manage those files, certainly for someone who is accessing the field programmable gate array(FPGA) for the first time.

As to claim 25, Muhlestein discloses an article of manufacture, comprising a processor-readable medium(memory 204, see Fig. 2) configured with instructions for causing a processor(processor 202, see Fig. 2) to perform the steps including, maintaining a hierarchy of directories and files(hierarchy of directories and files; see paragraph 0060, lines 3-4) in a virtual file system(virtual filer(VF2), see Fig. 4) that is registered with an operating system(storage operating system 300, see Fig. 4), wherein the directories and files are associated with resources of a programmable integrated circuit(IC)(field programmable gate array(FPGA); see paragraph 0051, lines 7-8)(see paragraph 0060, lines 1-7); in response to program calls(incoming request; see paragraph 0090, lines 6-7) to file system routines that reference files associated with resources(storage resource; see paragraph 0096, line 6) of the programmable IC, invoking the virtual file system(see paragraph 0096, lines 3-5 and Step 818, see Fig. 8); and accessing state information(configuration information stored in an IPSpace database; see paragraph 0090, lines17-18) in resources of the programmable IC by the virtual file system(see paragraph 0096, lines 5-6 and Steps 820 and 830, see Fig. 8).

However, Muhlestein fails to specifically disclose the article of manufacture wherein the directories and files have names indicative of resources of the programmable IC.

Bhesania teaches directories and files(information files; see paragraph 0015, lines 10-11) having names indicative of resources(see paragraph 0015, lines 10-12). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Muhlestein's and Bhesania's inventions in order to name Muhlestein's hierarchy of directories and files according to their resources. The motivation to do so would have been to better manage those files, certainly for someone who is accessing the field programmable gate array(FPGA) for the first time.

As to claim 26, Muhlestein discloses a system(filer 200, see Fig. 2) for providing access to configurable logic resources(storage resource; see paragraph 0096, line 6) on an integrated circuit (IC) (field programmable gate array(FPGA); see paragraph 0051, lines 7-8), comprising a processor(processor 202, see Fig. 2) hosting an operating system(storage operating system 300, see Fig. 2), the operating system accessing a hierarchy of directories and files(hierarchy of directories and files; see paragraph 0060, lines 3-4) in a virtual file system(virtual filer(VF2), see Fig. 4), wherein the directories and files are associated with the configurable logic resources(see paragraph 0060, lines 1-7); and code stored in a computer readable memory(memory 204, see Fig. 2), the code having program calls(incoming request; see paragraph 0090, lines 6-7) to file system routines that reference files associated with the configurable logic resources(storage resource; see paragraph 0096, line 6) such that the virtual file system is invoked(see paragraph 0096, lines 3-5 and Step 818, see Fig. 8) and state

information(configuration information stored in an IPSpace database; see paragraph 0090, lines 17-18) in the configurable logic resources is accessed by the virtual file system(see paragraph 0096, lines 5-6 and Steps 820 and 830, see Fig. 8). However, Muhlestein fails to specifically disclose the system wherein the directories and files have names indicative of resources of the programmable IC.

Bhesania teaches directories and files(information files; see paragraph 0015, lines 10-11) having names indicative of resources(see paragraph 0015, lines 10-12). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Muhlestein's and Bhesania's inventions in order to name Muhlestein's hierarchy of directories and files according to their resources. The motivation to do so would have been to better manage those files, certainly for someone who is accessing the field programmable gate array(FPGA) for the first time.

As to claim 27, Muhlestein discloses the system of claim 26 wherein the IC comprises an FPGA(field programmable gate array(FPGA); see paragraph 0051, lines 7-8).

As to claim 28, Muhlestein discloses the system of claim 26 wherein the processor is an embedded processor on the IC(see Fig. 2).

As to claim 29, Muhlestein discloses the system of claim 26 wherein the processor is a processor external to the IC(see Fig. 2).

As to claim 30, Muhlestein discloses the method of claim 1, wherein the hierarchy of directories and files includes a directory with a plurality of region files, each

region file associated with unused resources of the programmable IC(see paragraph 0060).

Response to Arguments

3. Applicant's arguments, see Remarks, filed 5/8/2008, with respect to the rejection(s) of claim(s) 1-13, 16, and 18-30 under 35 U.S.C. 102(e) as being anticipated by Muhlestein et al.[Muhlestein](US PGPub 2003/0191810) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Muhlestein et al.[Muhlestein](US PGPub 2003/0191810) in view of Bhesania et al.[Bhesania](US PGPub 2002/0122056).

Conclusion

4. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Brown whose telephone number is (571)272-5932. The examiner can normally be reached Monday-Thursday from 7:00am-5:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rehana Perveen can be reached on (571)272-3676. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael J. Brown
Art Unit 2116

/Thuan N. Du/
Primary Examiner, Art Unit 2116